Contamination screening of parasitic worm genome assemblies

Matthew Berriman (mb4@sanger.ac.uk)
Berriman Lab Group, Sanger Institute
Avril Coghlan
Berriman Lab Group, Sanger Institute
Daria Gordon
Berriman Lab Group, Sanger Institute

Method Article

Keywords: Genome sequencing, assembly, parasites, contamination, contaminants, genomes, assemblies, parasitic worms, helminths, nematodes, flatworms, platyhelminths, Nematoda, Platyhelminthes

Posted Date: April 11th, 2018

DOI: https://doi.org/10.1038/protex.2018.038

Abstract

A major problem in whole genome sequencing of parasitic worm \(nematode and flatworm) species is that sequencing reads can be contaminated with those of other species, either arising from DNA of the host species, other species that are commensal in the host, or from laboratory contamination. Here we describe a computational protocol to identify and remove likely contaminant scaffolds from the initial genome assembly for a parasitic worm. This protocol successfully identifies large contaminant scaffolds that contain many protein-coding genes.

Introduction

A major problem in whole genome sequencing of parasitic worm \(nematode and flatworm) species is that sequencing reads can be contaminated with those of other species, either arising from DNA of the host species \(e.g. vertebrates, plants, etc.), other species that are commensal in the host \(e.g. bacteria), or from laboratory contamination. Here we describe a computational protocol to identify and remove likely contaminant DNA from the initial genome assembly for a parasitic worm. See figure in Figures section. To remove contaminant scaffolds from the initial genome assembly for a parasitic worm \(nematode or flatworm), a multi-step approach is taken \(Figure 1). In Step A, we take each scaffold \(or 50 kb chunks of longer scaffolds) and run BLASTX \(Altschul et al 1997) against databases of invertebrate and non-invertebrate protein sequences. If a scaffold has far stronger BLASTX hits to non-invertebrate proteins \(e.g. vertebrates, bacteria) than to invertebrate proteins, it is considered to be a likely contaminant scaffold and removed. The second step, Step B, requires a gene set for the assembly, and runs BLASTP between the predicted proteins for this gene set and the same databases searched in Step A. As in Step A, scaffolds with far stronger BLASTP hits to non-invertebrate proteins are considered to be likely contaminants and are removed. Step B often detects additional contaminant scaffolds missed by Step A. Step C is designed to detect contamination of the parasitic worm's assembly by other invertebrates \(e.g. flatworm contamination in a nematode species' assembly). It is similar to Step B, but carries out additional BLASTP searches of a database of nematode or flatworm protein sequences. If a scaffold has far stronger BLASTP hits to non-invertebrate proteins from another phylum \(e.g. to flatworms, if the assembly being de-contaminated is from a nematode), then it is considered a likely contaminant and removed. Helminth genomes can be very large \(e.g. Fasciola hepatica ~1.3 Gb; Cwiklinski _et al_, 2015), so this approach is designed to be easy-to-run and scalable in terms of run-time to a large number of large parasitic worm genomes, with little or no manual analysis required. Our approach is designed to have few false positives \(non-contaminant scaffolds misclassified as contaminant). Our contamination scan protocol is designed particularly for parasitic worm genome assemblies, and relies on a series of BLASTX and BLASTP searches against invertebrate and non-invertebrate sequence databases. In contrast, some other approaches for contamination scanning can be used across a larger taxonomic breadth, and use additional data as well as BLAST searches. For example, Blobology \(Kumar _et al_ 2013), although designed with nematode genomes as a test case, can be used for any eukaryotic genome assembly, and analyses top BLAST hits but also the proportion of

GC bases and read coverage to identify likely contaminant scaffolds. Different contamination scan approaches likely disagree with respect to their verdict on some scaffolds. However, this may not matter much to the user in the case of very small scaffolds that lack any predicted genes. In contrast, missing a true contaminant scaffold that contains many protein-coding genes can have a large effect on downstream analyses \(e.g. of orthology), so we suggest that users may like to try both our protocol and others \(e.g. Blobology) to check if any additional large putative contaminant scaffolds are identified by one approach but not another. Such putative contaminants can then be subjected to manual scrutiny before deciding whether to discard them from a genome assembly.

Reagents

GenBank database

Equipment

Computer cluster.

Procedure

Step A: removing non-invertebrate contamination based on BLASTX hits The input to Step A is an initial genome assembly for a parasitic worm. 1. Each scaffold of the initial genome assembly is split into 50 kb chunks. 2. For each 50 kb chunk, BLASTX is run against two in-house sequence databases consisting of \(i) all invertebrate proteins from GenBank, and \(ii) all proteins in the full proteomes of representative species from major non-invertebrate taxa \(bacteria, vertebrates, fungi, plants, etc.), respectively. Only representative species are included here to reduce run-time. The –dbsize BLAST option is used to ensure that the E values from searching databases of different sizes are comparable. 3. For a particular chunk, if the e-value for its top non-invertebrate hit is 1E+10 fold lower than the e-value of its top invertebrate hit \(e.g. E-60 versus E-50), the chunk is considered to be contaminant. 4. If more than half of the chunks of a scaffold are classified as contaminant, the whole scaffold is considered contaminant and is removed from the assembly. The output from Step A is a genome assembly from which likely contaminant scaffolds have been removed. **Step B: removing non-invertebrate contamination based on BLASTP hits** The input to Step B is the de-contaminated genome assembly from Step A above, and a set of protein-coding gene predictions for this assembly \(e.g. from a gene-finding software such as Augustus \(Hoff & Stanke 2013)), along with their predicted protein sequences. 1. BLASTP searches \(using the –dbsize option) of predicted proteins from genes on scaffolds remaining after Step A are run against the search databases listed in Step A above. 2. For each protein, if its top BLASTP hit is to a non-invertebrate protein, and has an e-value that is 1E+50 times lower than that of the best invertebrate hit, then the gene is considered a putative contaminant gene. 3. Conversely, if the top hit is to an invertebrate protein, and its e-value is 1E+50 times lower than that of the best non-invertebrate hit, the gene is classified as non-contaminant. 4. If more than half of the classified genes on a scaffold are considered contaminant, then the scaffold is classified as contaminant and removed from the assembly.

The output from Step B is a genome assembly from which any additional likely contaminant scaffolds have been removed. **Step C: removing contamination from other invertebrates, based on BLASTP hits** Step C is a more stringent version of Step B, designed to remove contamination originating from other invertebrates \(for example, flatworm contamination in a nematode assembly), as well as any residual contamination from non-invertebrates \(e.g. bacteria) not removed by Steps A or B. The input to Step C is the de-contaminated genome assembly from Step B above, and a set of protein-coding gene predictions for this assembly. 1. For the non-contaminant scaffolds that remain after step B, predicted protein sequences for genes on these scaffolds are BLASTP-searched \(with the –dbsize option) against the database of non-invertebrate proteins used in steps A and B, plus either nematode or flatworm protein sequences from GenBank. 2. For each query gene on a scaffold from a flatworm species' assembly, the top ten BLASTP hits in nematodes/non-invertebrates and in the flatworm database are recorded. 3. If the top five of these ten BLAST hits are to nematodes/non-invertebrates, and the e-value of the worst nematode/non-invertebrate hit is at least 5 orders of magnitude lower than the e-value of the best flatworm hit, the query gene is considered to be a contaminant gene. 4. Conversely, if the top five of the ten hits are to flatworm, and the e-value of the worst flatworm hit is 5 orders of magnitude lower than that of the best nematode/non-invertebrate hit, the query gene is considered a non-contaminant gene. 5. If a scaffold has one or more contaminant genes, and no non-contaminant genes, it is considered to be a contaminant scaffold and removed. The output from Step C is a genome assembly from which any additional likely contaminant scaffolds have been removed.

Troubleshooting

Based on manual analysis of the results for some species of interest, we find this approach successfully removes most large contaminant scaffolds from a parasitic worm genome assembly. The protocol has been designed to have a low rate of false positives \(i.e. it should classify few non-contaminant scaffolds as contaminants). In support of this, we find that it classifies few or no scaffolds as contaminant in genome assemblies that have undergone extensive manual improvement \(e.g. _Strongyloides ratti_; Hunt _et al_ 2017). However, the low rate of false positives is at the expense of some false negatives: after de-contamination, an assembly may still contain some small \(often <10 kb) contaminant scaffolds that are hard to detect. In particular, our approach is largely based on finding contaminant genes, so may miss contaminant scaffolds that lack coding sequences. Furthermore, since Step B uses a database of proteomes from certain representative non-invertebrate species \(e.g. _Escherichia coli_ K12 rather than every _E. coli_ strain), it is possible \(although probably relatively rare) that it will miss a contaminant scaffold originating from a different _E. coli_ strain that contains mostly genes missing from _E. coli_ K12. It is likely that our approach has few false positives, as long as the query assembly is from a parasitic worm clade for which many species have previously been sequenced \(e.g., nematode clade V). However, Step C may give rise to some false positives if the query assembly is from a clade that is poorly represented in the NCBI database. For example, if the query assembly is for a clade I nematode species, it is possible that some diverged clade I genes may have slightly stronger BLAST hits to flatworms than to other previously sequenced nematodes, and so may be misclassified as contaminants by Step C. If the

user is concerned about the possibility of false positives from Step C \(e.g. if their query genome is from a parasitic clade for which few species have been sequenced before), we suggest that contaminant scaffolds identified by Step C should be manually examined, and added back to the assembly if doubt remains after manual analysis.

Anticipated Results

The output from the protocol is a genome assembly from which any likely contaminant scaffolds have been removed.

References

Altschul, S.F. _et al._ Gapped BLAST and PSI-BLAST: a new generation of protein database search programs. _Nucleic Acids Res._ **25**, 3389-3402 \(1997). Cwiklinski, K. _et al._ The _Fasciola hepatica_ genome: gene duplication and polymorphism reveals adaptation to the host environment and the capacity for rapid evolution. _Genome Biol._ **16**, 71 \(2015). Hoff, K.J. & Stanke, M. WebAUGUSTUS–a web service for training AUGUSTUS and predicting genes in eukaryotes. _Nucleic Acids Res._ **41**, W123-128 \(2013). Hunt, V.L. _et al._ The genomic basis of parasitism in the _Strongyloides_ clade of nematodes. _Nat Genet._ **48**, 299-307 \(2016). Kumar, S. _et al._ Blobology: exploring raw genome data for contaminants, symbionts and parasites using taxon-annotated GC-coverage plots. _Front Genet._ **4**, 237 \(2013).

Acknowledgements

We would like to thank the WTSI Pathogen Informatics team, especially Jacqueline Keane and Andrew Page.

Figures

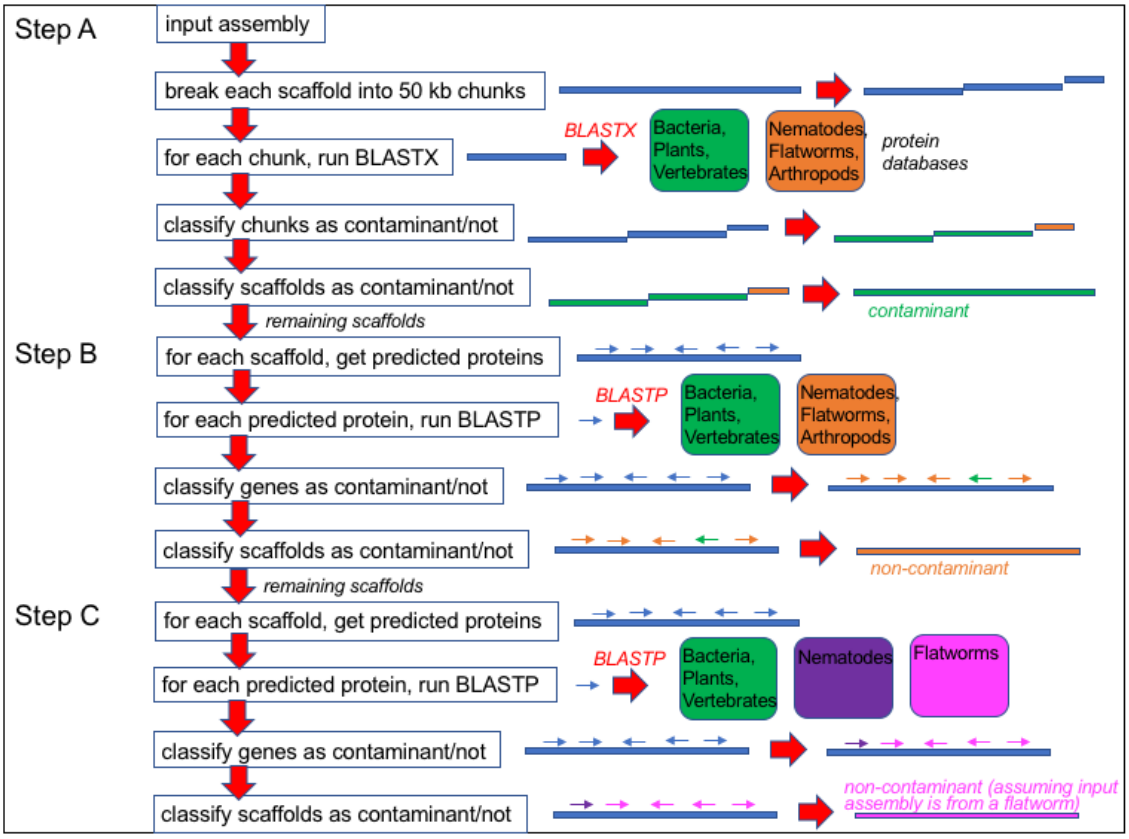

Figure 1

Flowchart of protocol